Amendments to the claims:

Please amend the claims as follows:

1. (Currently Amended) A method for deriving transformations for transforming data from one data schema to another, comprising:

receiving a source data schema and a target data schema, the source data schema being different from the target data schema, each of the source data schema and the target data schema defining data on a computer readable media;

mapping the source data schema into an ontology model;

mapping the target data schema into the ontology model; and

automatically deriving, using the ontology model, a transformation for transforming data conforming to the source data schema into data conforming to the target data schema, ~~using the ontology model~~

wherein the automatically derived transformation transforms data conforming to the source data schema directly to data conforming to the target data schema.

2. (Original) The method of claim 1 further comprising converting at least one of the source data schema and the target schema from an external format to an internal format.

3. (Original) The method of claim 1 further comprising receiving the ontology model.

4. (Original) The method of claim 3 further comprising converting the ontology model from an external format to an internal format.

5. (Original) The method of claim 1 further comprising generating the ontology model.

6. (Original) The method of claim 5 further comprising receiving an initial ontology model, wherein said generating generates the ontology model from the initial ontology model.

7. (Original) The method of claim 6 further comprising converting the initial ontology model from an external format to an internal format.

8. (Currently Amended) The method of claim 1 further comprising automatically generating executable program code that transforms data conforming to the source data schema into data conforming to the target data schema.

9. (Original) The method of claim 1 wherein the source data schema is a source table schema describing source data tables, wherein the target data schema is a target table schema describing target data tables, and wherein the source table schema and the target table schema each describes at least one table having columns.

10. (Currently Amended) The method of claim 9 wherein the source table schema is a source relational database schema describing source relational database tables, wherein the target table schema is a target relational database schema describing target relational database tables, and wherein the automatically derived transformation is an SQL query.

11. (Original) The method of claim 10 wherein said mapping a source data schema and said mapping a target data schema each comprise:

identifying at least one class in the ontology model corresponding to at least one table; and

identifying at least one property or composition of properties in the ontology model corresponding to at least one table column.

12. (Currently Amended) The method of claim 11 wherein said automatically deriving comprises:

labeling properties of the ontology model with symbols;

converting at least one column in the source relational database schema into at least one source symbol;

converting at least one column in the target relational database schema into at least one target symbol; and

expressing the at least one target symbol in terms of at least one source symbol.

13. (Original) The method of claim 12 wherein said expressing uses expressions involving composition of properties.

14. (Currently Amended) The method of claim 12 wherein at least one dependency exists among properties in the ontology model, and wherein said automatically deriving further comprises translating the at least one dependency among properties in the ontology model as at least one dependency between target relational database columns and source relational database columns, and wherein said expressing incorporates the at least one dependency between target relational database columns and source relational database columns.

15. (Original) The method of claim 14 wherein said expressing uses expressions involving arithmetic operations.

16. (Original) The method of claim 14 wherein said expressing uses expressions involving character string operations.

17. (Original) The method of claim 10 further comprising applying the query to at least one source relational database table to populate at least one target relational database table.

18. (Original) The method of claim 17 wherein the at least one source relational database table reside in a single database.

19. (Original) The method of claim 17 wherein the at least one source relational database table reside in multiple databases.

20. (Original) The method of claim 1 wherein the source data schema is a source document schema describing source documents, and wherein the target data schema is a target document schema describing target documents.

21. (Original) The method of claim 20 wherein the source document schema is a source DTD describing source XML documents, wherein the target document schema is a target DTD describing target XML documents, and wherein the source DTD and the target DTD each describes at least one XML element or XML attribute.

22. (Original) The method of claim 21 wherein the transformation is an XQuery.

23. (Original) The method of claim 21 wherein the transformation is an XSLT script.

24. (Original) The method of claim 20 wherein the source document schema is a source XML schema describing source XML documents, wherein the target document schema is a target XML schema describing target XML documents, and wherein the source XML schema and the target XML schema each describes at least one XML complexType having at least one XML element or XML attribute.

25. (Original) The method of claim 24 wherein the transformation is an XQuery.

26. (Original) The method of claim 24 wherein the transformation is an XSLT script.

27. (Original) The method of claim 24 wherein said mapping a source data schema and said mapping a target data schema each comprise:

identifying at least one class in the ontology model corresponding to at least one XML complexType; and

identifying at least one property or composition of properties in the ontology model corresponding to at least one XML element or XML attribute.

28. (Currently Amended) The method of claim 24 wherein said automatically deriving comprises expressing XML elements and XML attributes of the target XML schema in terms of XML elements and XML attributes of the source XML schema.

29. (Original) The method of claim 28 wherein said expressing is performed recursively through XPath paths.

30. (Currently Amended) The method of claim 27 wherein at least one dependency exists among properties in the ontology model, and wherein said automatically deriving further comprises translating the at least one dependency among properties in the ontology model as at least one dependency between target XML elements and source XML elements.

31. (Original) The method of claim 26 further comprising applying the XSLT script to at least one source XML document to generate at least one target XML document.

32. (Original) The method of claim 31 wherein the at least one source XML document reside in a single database.

33. (Original) The method of claim 31 wherein the at least one source XML document reside in multiple databases.

34. (Currently Amended) A system for deriving transformations for transforming data from one data schema to another, comprising:

a schema receiver receiving a source data schema and a target data schema the source data schema being different than the target data schema;

a mapping processor mapping a data schema into an ontology model; and

a transformation processor deriving a transformation for transforming data conforming to the source data schema into data conforming to the target data schema, based on respective source and target mappings generated by said mapping processor

for mapping said source data schema and said target data schema into a common ontology model,

wherein the derived transformation transforms data conforming to the source data schema directly into data conforming to the target data schema.

35. (Currently Amended) The system of claim 34 further comprising a schema format converter ~~convertor~~, converting at least one of the source data schema and the target data schema from an external format to an internal format.

36. (Original) The system of claim 34 further comprising an ontology receiver receiving the ontology model.

37. (Currently Amended) The system of claim 36 further comprising an ontology format converter ~~convertor~~, converting the ontology model from an external format to an internal format.

38. (Original) The system of claim 34 further comprising an ontology builder generating the ontology model.

39. (Original) The system of claim 38 further comprising an ontology receiver receiving an initial ontology model, wherein said ontology builder generates the ontology model from the initial ontology model.

40. (Currently Amended) The system of claim 39 further comprising an ontology format converter ~~convertor~~, converting the initial ontology model from an external format to an internal format.

41. (Currently Amended) The system of claim 34 further comprising a program code generator generating executable program code operable to perform the derived transformation ~~that transforms data conforming to the source data schema into data conforming to the target data schema~~.

42. (Original) The system of claim 34 wherein the source data schema is a source table schema describing source data tables, wherein the target data schema is a target table schema describing target data tables, and wherein the source table schema and the target table schema each describes at least one data table having columns.

43. (Original) The system of claim 42 wherein the source table schema is a source relational database schema describing source relational database tables, wherein the target table schema is a target relational database schema describing target database tables, and wherein the transformation is an SQL query.

44. (Original) The system of claim 43 wherein said mapping processor comprises:

a class identifier identifying at least one class in the common ontology model corresponding to at least one table; and

a property identifier identifying at least one property or composition of properties in the common ontology model corresponding to at least one table column.

45. (Original) The system of claim 44 wherein said property identifier presents a user with a choice of at least one property in the common ontology model that may correspond to a given table column.

46. (Original) The system of claim 45 wherein the choice of at least one property only includes properties having targets that are compatible with a data type of the given table column.

47. (Original) The system of claim 46 wherein, for a given table column that is a foreign key to a foreign table, the choice of at least one property only includes properties whose target is a class corresponding to the foreign table.

48. (Original) The system of claim 43 wherein said transformation processor comprises:

an ontology labeller labeling properties of the common ontology model with symbols;

a column converter converting at least one column in the source relational database schema into at least one source symbol, and converting at least one column in the target relational database schema into at least one target symbol; and

a symbol processor expressing the at least one target symbol in terms of at least one source symbol.

49. (Original) The system of claim 48 wherein said symbol processor uses expressions involving composition of properties.

50. (Original) The system of claim 48 wherein at least one dependency exists among properties in the ontology model, and wherein said transformation processor further comprises a dependency processor translating the at least one dependency among properties in the ontology model as at least one dependency between target relational database columns and source relational database columns, and wherein said symbol processor incorporates the at least one dependency between target relational database columns and source relational database columns.

51. (Original) The system of claim 50 wherein said symbol processor uses expressions involving arithmetic operations.

52. (Original) The system of claim 50 wherein said symbol processor uses expressions involving character string operations.

53. (Original) The system of claim 43 further comprising:

a data receiver receiving at least one source relational database table; and

a data processor applying the query to the at least one source relational database table to populate at least one target relational database table.

54. (Original) The system of claim 53 wherein the at least one source relational database table reside in a single database.

55. (Original) The system of claim 53 wherein the at least one source relational database table resides in multiple databases.

56. (Original) The system of claim 34 wherein the source data schema comprises a source document schema describing source documents, and wherein the target data schema comprises a target document schema describing target documents.

57. (Original) The system of claim 56 wherein the source document schema is a source DTD describing source XML documents, wherein the target document schema is a target DTD describing target XML documents, and wherein the source DTD and the target DTD each describes at least one XML element or XML attribute.

58. (Currently Amended) The system of claim 57 wherein the derived transformation is an XQuery.

59. (Currently Amended) The system of claim 57 wherein the derived transformation is an XSLT script.

60. (Original) The system of claim 56 wherein the source document schema comprises a source XML schema that describes XML source documents, wherein the target document schema comprises a target XML schema that describes XML target documents, and wherein the source XML schema and the target XML schema each comprises at least one XML complexType having at least one XML element or XML attribute.

61. (Currently Amended) The system of claim 60 wherein the derived transformation is an XQuery.

62. (Currently Amended) The system of claim 60 wherein the derived transformation is an XSLT script.

63. (Original) The system of claim 60 wherein said mapping processor comprises:

a class identifier identifying at least one class in the ontology model corresponding to at least one XML complexType; and

an property identifier identifying at least one property or composition of properties in the ontology model corresponding to at least one XML element or XML attribute.

64. (Original) The system of claim 60 wherein said transformation processor comprises an XML processor expressing XML elements and XML attributes of said target XML schema in terms of XML elements and XML attributes of said source XML schema.

65. (Original) The system of claim 64 wherein said XML processor operates recursively through XPath paths.

66. (Original) The system of claim 64 wherein at least one dependency exists among properties in the ontology model, and wherein said transformation processor further comprises a dependency processor translating the at least one dependency among properties in the ontology model as at least one dependency between target XML elements or attributes, and source XML elements or attributes, and wherein said XML processor incorporates the at least one dependency between target XML elements or attributes, and source XML elements or attributes.

67. (Original) The system of claim 60 further comprising

a data receiver receiving at least one source XML document; and

a data processor applying the XSLT script to the at least one source XML document to generate at least one target XML document.

68. (Original) The system of claim 67 wherein the at least one source XML document reside in a single database.

69. (Original) The system of claim 67 wherein the at least one source XML document reside in multiple databases.

70. (Currently Amended) A method for building an ontology model into which data schema can be embedded, comprising:

receiving at least one data schema defining data on a computer readable media; and

building an ontology model at least partially based on components of the received at least one data schema into which the at least one data schema can be embedded, the building being at least partially automatic.

71. (Original) The method of claim 70 further comprising converting at least one of the at least one data schema from an external format to an internal format.

72. (Original) The method of claim 70 wherein the at least one data schema is at least one table schema describing data tables having columns.

73. (Original) The method of claim 72 wherein the at least one table schema is at least one relational database schema describing relational database tables.

74. (Original) The method of claim 73 wherein said building an ontology model comprises:

providing an initial ontology model;

adding classes to the initial ontology model corresponding to tables described in the at least one relational database schema; and

adding properties to the initial ontology model corresponding to columns described in the at least one relational database schema.

75. (Original) The method of claim 74 wherein the initial ontology model is empty.

76. (Original) The method of claim 74 wherein the initial ontology model is non-empty.

77. (Original) The method of claim 76 further comprising converting the initial ontology model from an external format to an internal format.

78. (Original) The method of claim 74 wherein said adding classes is performed by a computer in conjunction with a user.

79. (Original) The method of claim 78 wherein said adding classes prompts a user to add a class to the ontology model when there is a table described in the at least one relational database schema that does not correspond to an existing class in the ontology model.

80. (Original) The method of claim 74 wherein said adding classes is performed automatically by a computer.

81. (Original) The method of claim 80 wherein said adding classes automatically adds a class to the ontology model when there is a table described in the at least one relational database schema that does not correspond to an existing class in the ontology model.

82. (Original) The method of claim 74 wherein said adding properties is performed by a computer in conjunction with a user.

83. (Original) The method of claim 82 wherein said adding properties prompts a user to add a property to the ontology model when there is a table column described in the at

least one relational database schema that does not correspond to an existing property or composition of properties in the ontology model.

84. (Original) The method of claim 74 wherein said adding properties is performed automatically by a computer.

85. (Original) The method of claim 84 wherein said adding properties automatically adds a property to the ontology model when there is a table column described in the at least one relational database schema that does not correspond to an existing property or composition of properties in the ontology model.

86. (Original) The method of claim 70 wherein said building an ontology model comprises inferring inheritance relationships between classes in the ontology model based on relationships between tables described in the at least one relational database schema.

87. (Original) The method of claim 86 wherein a first class in the ontology model is inferred to inherit from a second class in the ontology model when a table corresponding to the first class has a primary key that is a foreign key to a table corresponding to the second class.

88. (Original) The method of claim 86 wherein said inferring inheritance relationships includes prompting a user to confirm an inferred inheritance relationship.

89. (Original) The method of claim 70 wherein the at least one data schema is at least one document schema describing documents.

90. (Original) The method of claim 89 wherein the at least one document schema is an XML schema describing XML documents having at least one XML complexType with at least one XML element or XML attribute.

91. (Original) The method of claim 90 wherein said building an ontology model comprises:

providing an initial ontology model;

adding classes to the initial ontology model corresponding to XML complexTypes described in the at least one XML schema; and

adding properties to the initial ontology model corresponding to XML elements and XML attributes described in the at least one XML schema.

92. (Original) The method of claim 91 wherein the initial ontology model is empty.

93. (Original) The method of claim 92 wherein the initial ontology model is non-empty.

94. (Original) The method of claim 91 wherein said adding classes is performed by a computer in conjunction with a user.

95. (Original) The method of claim 94 wherein said adding classes prompts a user to add a class to the ontology model when there is an XML complexType described in the at least one XML schema that does not correspond to an existing class in the ontology model.

96. (Original) The method of claim 91 wherein said adding classes is performed automatically by a computer.

97. (Original) The method of claim 96 wherein said adding classes automatically adds a class to the ontology model when there is an XML complexType described in the at least one XML schema that does not correspond to an existing class in the ontology model.

98. (Original) The method of claim 91 wherein said adding properties is performed by a computer in conjunction with a user.

99. (Original) The method of claim 98 wherein said adding properties prompts a user to add a property to the ontology model when there is an XML element or an XML attribute described in the at least one XML schema that does not correspond to an existing property or composition of properties in the ontology model.

100. (Original) The method of claim 91 wherein said adding properties is performed automatically by a computer.

101. (Original) The method of claim 100 wherein said adding properties automatically adds a property to the ontology model when there is an XML element or an XML attribute described in the at least one relational database schema that does not correspond to an existing property or composition of properties in the ontology model.

102. (Currently Amended) A system for building an ontology model into which data schema can be embedded, comprising:

a schema receiver receiving at least one data schema defining data on a computer readable media; and

a model builder building an ontology model at least partially based on components of the received at least one data schema into which the at least one data schema can be embedded.

103. (Currently Amended) The system of claim 102 further comprising a schema format converter ~~converter~~, converting at least one of the at least one data schema from an external format to an internal format.

104. (Original) The system of claim 102 wherein the at least one data schema is at least one table schema describing data tables having columns.

105. (Original) The system of claim 104 wherein the at least one table schema is at least one relational database schema describing relational database tables.

106. (Original) The system of claim 105 further comprising an ontology receiver receiving an initial ontology model, and wherein said model builder comprises:

a class adder adding classes to the initial ontology model corresponding to tables described in the at least one relational database schema; and

a property adder adding properties to the initial ontology model corresponding to table columns described in the at least one relational database schema.

107. (Original) The system of claim 106 wherein the initial ontology model is empty.

108. (Original) The system of claim 106 wherein the initial ontology model is non-empty.

109. (Currently Amended) The system of claim 108 further comprising an ontology format converter ~~convertor~~, converting the initial ontology model from an external format to an internal format.

110. (Original) The system of claim 106 wherein said class adder is guided by a user in conjunction with a computer.

111. (Original) The system of claim 110 wherein said class adder prompts a user to add a class to the ontology model when there is a table described in the at least one relational database schema that does not correspond to an existing class in the ontology model.

112. (Original) The system of claim 106 wherein said class adder is automatically guided by a computer.

113. (Original) The system of claim 112 wherein said class adder automatically adds a class to the ontology model when there is a table described in the at least one relational database schema that does not correspond to an existing class in the ontology model.

114. (Original) The system of claim 106 wherein said property adder is guided by a user in conjunction with a computer.

115. (Original) The system of claim 114 wherein said property adder prompts a user to add a property to the ontology model when there is a table column described in the at least one relational database schema that does not correspond to an existing property or composition of properties in the ontology model.

116. (Original) The system of claim 106 wherein said property adder is automatically guided by a computer.

117. (Original) The system of claim 116 wherein said property adder automatically adds a property to the ontology model when there is a table column described in the at least one relational database schema that does not correspond to an existing property or composition of properties in the ontology model.

118. (Original) The system of claim 105 wherein said model builder comprises an inheritance processor inferring inheritance relationships between classes in the ontology model based on relationships between tables in the at least one relational database schema.

119. (Original) The system of claim 118 wherein said inheritance processor infers that a first class in the ontology model inherits from a second class in the ontology model when a table corresponding to the first class has a primary key that is a foreign key to a table corresponding to the second class.

120. (Original) The system of claim 118 wherein said model builder ensures that classes corresponding to tables in the at least one relational database schema obey the inferred inheritance relationships.

121. (Original) The system of claim 120 wherein said inheritance processor prompts a user to confirm an inferred inheritance relationship.

122. (Original) The system of claim 102 wherein the at least one data schema comprises at least one document schema describing documents.

123. (Original) The system of claim 122 wherein the at least one document schema comprises at least one XML schema that describes XML documents, wherein having at least one XML complexType with at least one XML element or XML attribute.

124. (Original) The system of claim 123 further comprising an ontology receiver receiving an initial ontology model, and wherein said model builder comprises:

a class adder adding classes to the initial ontology model corresponding to XML complexTypes described in the at least one XML schema; and

a property adder adding properties to the initial ontology model corresponding to table columns in the at least one relational database schema.

125. (Original) The system of claim 124 wherein the initial ontology model is empty.

126. (Original) The system of claim 124 wherein the initial ontology model is non-empty.

127. (Original) The system of claim 124 wherein said class adder is guided by a user in conjunction with a computer.

128. (Original) The system of claim 127 wherein said class adder prompts a user to add a class to the ontology model when there is an XML complexType described in the at least one XML schema that does not correspond to an existing class in the ontology model.

129. (Original) The system of claim 124 wherein said class adder is automatically guided by a computer.

130. (Original) The system of claim 129 wherein said class adder automatically adds a class to the ontology model when there is an XML complexType described in the at least one XML schema that does not correspond to an existing class in the ontology model.

131. (Original) The system of claim 124 wherein said property adder is guided by a user in conjunction with a computer.

132. (Original) The system of claim 131 wherein said property adder prompts a user to add a property to the ontology model when there is an XML element or XML attribute described in the at least one XML schema that does not correspond to an existing property or composition of properties in the ontology model.

133. (Original) The system of claim 124 wherein said property adder is automatically guided by a computer.

134. (Original) The system of claim 133 wherein said property adder automatically adds a property to the ontology model when there is an XML element or XML attribute described in the at least one XML schema that does not correspond to an existing property or composition of properties in the ontology model.

135. (Currently Amended) An article of manufacture including one or more computer-readable media that embody a program of instructions for transforming data from one schema to another, wherein the program of instructions, when executed by a processing system, causes the processing system to:

receive a source data schema and a target data schema, the source data schema being different that the target data schema;

map the source data schema into an ontology model;

map the target data schema into the ontology model; and

derive a transformation for transforming data conforming to the source data schema into data conforming to the target relational database schema, using the ontology model,

wherein the derived transformation transforms data conforming to the source data schema directly into data conforming to the target data schema.

136. (Original) The article of claim 135 wherein the one or more computer-readable media include one or more non-volatile storage devices.

137. (Cancelled)

138. (Currently Amended) An article of manufacture including one or more computer-readable media that embody a program of instructions for building a common ontology model into which data schema can be embedded, wherein the program of instructions, when executed by a processing system, causes the processing system to:

receive at least one data schema; and

build an ontology model at least partially based on said received at least one data schema into which the at least one data schema can be embedded.

139. (Original) The article of claim 138 wherein the one or more computer-readable media include one or more non-volatile storage devices.

140. Cancelled